Unsupervised Estimation for Noisy-Channel Models

Markos Mylonakis MMYLONAK@SCIENCE.UVA.NL
Khalil Sima'an SIMAAN@SCIENCE.UVA.NL
Language and Computation, University of Amsterdam, Pl. Muidergracht 24, 1018TV Amsterdam, Netherlands

Rebecca Hwa HWA@CS.PITT.EDU
Departement of Computer Science, University of Pittsburgh, 210 S. Bouquet St. Pittsburgh, PA 15260, U.S.A.

Abstract

Shannon's Noisy-Channel model, which describes how a corrupted message might be reconstructed, has been the corner stone for much work in statistical language and speech processing. The model factors into two components: a *language model* to characterize the original message and a *channel model* to describe the channel's corruptive process. The standard approach for estimating the parameters of the channel model is unsupervised Maximum-Likelihood of the observation data, usually approximated using the Expectation-Maximization (EM) algorithm. In this paper we show that it is better to maximize the joint likelihood of the data *at both ends of the noisy-channel*. We derive a corresponding bi-directional EM algorithm and show that it gives better performance than standard EM on two tasks: (1) translation using a probabilistic lexicon and (2) adaptation of a part-of-speech tagger between related languages.

1. Introduction

An influential paradigm in statistical natural language processing (NLP) is the noisy-channel model (Shannon & Weaver, 1949). It describes a communication process in which a sender emits the intended message $\mathbf{m}$ through an imperfect communication channel such that the sequence $\mathbf{o}$ observed by the recipient is a "noisy" version of the original message. To reconstruct $\mathbf{m}$ from $\mathbf{o}$, one may postulate a set of hypotheses, $\mathcal{H}(\mathbf{o})$, and compute the optimal Bayesian hypothesis, $\mathbf{m}^* = \arg\max_{\mathbf{m}\in\mathcal{H}(\mathbf{o})} P(\mathbf{m} \mid \mathbf{o}) = \arg\max_{\mathbf{m}\in\mathcal{H}(\mathbf{o})} P(\mathbf{m})P(\mathbf{o} \mid \mathbf{m})$, where $P(\mathbf{m})$ is called the *language model* and $P(\mathbf{o} \mid \mathbf{m})$ the *channel model*. Many NLP problems can be framed in terms of the noisy-channel model. For example, in speech recognition, $\mathbf{o}$ is an acoustic utterance heard by the recipient and $\mathbf{m}$ is the speaker's intended message; in machine translation, $\mathbf{o}$ is a sentence expressed in a foreign language, $\mathbf{s}$ (source); $\mathbf{m}$ is the intended message expressed in the recipient's native language, $\mathbf{t}$ (target); and the channel model is a probabilistic translation lexicon (dictionary).

A major challenge for training the channel model for an NLP application is that the available data rarely contains explicit, in-depth mappings between $\mathbf{o}$ and $\mathbf{m}$. For instance, consider the problem of training a channel model for machine translation. While it may not be hard to find bilingual texts, the texts themselves do not specify how individual words in the source language are translated into words in the target language. Thus, the channel model $P(\mathbf{o} \mid \mathbf{m})$ is usually explained by assuming a distribution over a hidden "translation" relation, $\mathbf{a} \subseteq \mathbf{m}\times\mathbf{o}$, so that $P(\mathbf{o} \mid \mathbf{m}) = \sum_{\mathbf{a}} P(\mathbf{o}, \mathbf{a} \mid \mathbf{m})$ (Bahl et al., 1990; Brown et al., 1988). The parameters for the model can be estimated with the Expectation-Maximization (EM) algorithm (Baum et al., 1970; Dempster et al., 1977). However, this means that the parameters are fitted only to data from one side of the channel: The language model parameters depend solely on data from the message-side; and the channel model parameters are chosen to maximize the likelihood of the data from the observable-side of the channel alone. Because of weak language models, asymmetric channel models and sparse-data, this approach leads to different estimates from each direction of the channel ($P(\mathbf{m})P(\mathbf{o}|\mathbf{m})$ vs. $P(\mathbf{o})P(\mathbf{m}|\mathbf{o})$). Some recent work (Zens et al., 2004; Liang et al., 2006) suggests that this could be suboptimal in practice and that the two directions of the channel should be reconciled.

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

In this paper we explore methods of maximizing *the likelihood of both the observable and message sides* of the training data simultaneously. We propose that the two directions of translation, $P(\mathbf{o})\sum_{\mathbf{a}} P(\mathbf{a},\mathbf{m} \mid \mathbf{o})$ and $P(\mathbf{m})\sum_{\mathbf{a}} P(\mathbf{a},\mathbf{o} \mid \mathbf{m})$, employ the same set of joint probabilities $P(\mathbf{o},\mathbf{a},\mathbf{m})$. This allows training on the joint data of messages and observations under Maximum-Likelihood. We extend the standard EM algorithm into a Bi-directional EM (Bi-EM) algorithm for re-estimating channel parameters. Unlike standard NLP application of the noisy-channel model, our algorithm does not depend on using a parallel corpus of messages and their corresponding corrupted observations $(\mathbf{m},\mathbf{o})$ as the training data; it is sufficient to have separate corpora of $\mathbf{m}$ and $\mathbf{o}$. This is especially beneficial for machine translation between languages for which bilingual texts are not abundant.

We present experiments comparing Bi-EM with the uni-directional EM on two tasks (1) translation from one language to another using a probabilistic translation lexicon and two monolingual corpora, and (2) automatic adaptation of a part-of-speech (POS) tagger from a language for which there exists an annotated training corpus (written, Modern Standard Arabic) to a related language (Spoken, Levantine dialect) for which there is only a small, unannotated corpus of sentences. On both tasks, and under varying training conditions, the Bi-EM estimates give better system performance than standard (unidirectional) EM.

2. Background and Related Work

It is useful to think of the noisy-channel problem as a translation task: the observation $\mathbf{o}$ is the source language sentence $\mathbf{s}$ and the message $\mathbf{m}$ is the target language sentence $\mathbf{t}$. While channel models ($P(\mathbf{s} \mid \mathbf{t})$) can be implemented in many ways, in this paper we consider only a probabilistic translation lexicon that bridges the source (observation) and target (message) texts. This choice does not impact the generality of the estimation algorithm presented, especially with regard to applications such as machine translation or speech-recognition. Much work in Statistical Machine Translation (SMT) has been devoted to the estimation of lexicon probabilities. We briefly review the relevant literature as a background against which we present our algorithm.

2.1. Translation Probabilities in SMT

For a source sentence $\mathbf{s} = (s_1,\ldots,s_n)$ and a target sentence $\mathbf{t} = (t_1,\ldots t_m)$, the objective of SMT can be expressed in the noisy-channel framework as:

$$\arg\max_{\mathbf{t}} p(\mathbf{t}|\mathbf{s}) = \arg\max_{\mathbf{t}} p(\mathbf{s}|\mathbf{t})p(\mathbf{t}).$$

To learn the translation model, most SMT approaches require a large parallel corpus (see e.g. (Brown et al., 1988; Koehn et al., 2003)) in order to induce a hidden alignment $\mathbf{a}$ between the words of each pair of sentences $\mathbf{s}$ and $\mathbf{t}$:

$$\arg\max_{\mathbf{t}} p(\mathbf{t}|\mathbf{s}) = \arg\max_{\mathbf{t}} \sum_{\mathbf{a}} p(\mathbf{s},\mathbf{a}|\mathbf{t})p(\mathbf{t}).$$

To estimate the word alignment probabilities and the lexicon probabilities, most work employs some form of the Expectation-Maximization algorithm.

2.2. Baseline Model

In contrast with work using parallel corpora, in (Koehn & Knight, 2000) as well as in this paper, only monolingual corpora (in both source and target languages) are available. Because the two corpora are not translations of each other, alignments between the pairs of sentences by-and-large do not exist. Instead, we assume that we are provided with an ambiguous translation lexicon L (which may be obtained from a bilingual dictionary). For every source word s, L contains a set of translations $L(s)$, and vice versa (for target word t it contains a set $L(t)$). The goal is to estimate translation probabilities $p(s|t)$, the probability that a word t translates as word $s \in L(t)$, regardless of context. Let the set $L(\mathbf{s})$ stand for the set of all possible target sentences $\mathbf{t}$ that result from translating the (ordered) sequence of words in $\mathbf{s}$, one by one[1], using lexicon L. Koehn and Knight derive the following model[2]

$$\arg\max_{\mathbf{t}\in L(\mathbf{s})} p(\mathbf{t}|\mathbf{s}) = \arg\max_{\mathbf{t}\in L(\mathbf{s})} p_{\overleftarrow{\theta}}(\mathbf{s}|\mathbf{t})p(\mathbf{t}) \quad (1)$$

$$= \arg\max_{\mathbf{t}\in L(\mathbf{s})} p(\mathbf{t}) \prod_{i=1}^{n} \overleftarrow{\theta}(s_i|t_i) \quad (2)$$

where $\overleftarrow{\theta}$ stands for the translation lexicon probabilities $\mathbf{s} \leftarrow \mathbf{t}$, i.e. $p(\mathbf{s}|\mathbf{t})$. This model employs a language model $p(\mathbf{t})$ over target sentences trained on the target language monolingual corpus $\mathcal{T}$, and a "translation model" with lexicon probabilities $\overleftarrow{\theta}(s_i|t_i)$.

Using fixed language model estimates $\tilde{p}(\mathbf{t})$, the lexicon probabilities are estimated using EM over the source

[1] Thereby assuming the same word-order and a one-to-one mapping between words, which also implies that sentence length is unchanged, i.e. $m == n$.

[2] The notation $p_\theta(.)$ stands for the probability under model (parameters) θ.

language corpus $\mathcal{S}$. Assuming an initial estimate $\overleftarrow{\theta}_0$ for $\overleftarrow{\theta}$, and denote the estimate at iteration r by $\overleftarrow{\theta}_r$

E-step$_r$: for every $\mathbf{s} \in \mathcal{S}$ and $\mathbf{t} \in L(\mathbf{s})$:

$$q(\mathbf{t}|\mathbf{s}; \overleftarrow{\theta}_r) := \frac{1}{Z(\mathbf{s}; \overleftarrow{\theta}_r)} \tilde{p}(\mathbf{t}) \prod_{i=1}^{n} \overleftarrow{\theta}_r(s_i|t_i)$$

M-step$_r$: maximize over $\overleftarrow{\theta}$ to obtain $\overleftarrow{\theta}_{r+1}$

$$\overleftarrow{\theta}_{r+1} := \arg\max_{\overleftarrow{\theta}} \sum_{\substack{\mathbf{s}\in\mathcal{S} \\ \mathbf{t}\in L(\mathbf{s})}} q(\mathbf{t}|\mathbf{s}; \overleftarrow{\theta}_r) \log[\tilde{p}(\mathbf{t}) p_{\overleftarrow{\theta}}(\mathbf{s}|\mathbf{t})]$$

where $Z(\mathbf{s}; \overleftarrow{\theta}_r) = \sum_{\mathbf{t}\in L(\mathbf{s})} \tilde{p}(\mathbf{t}) \prod_{i=1}^{n} \overleftarrow{\theta}_r(s_i|t_i)$. The maximization at iteration r (M-step$_r$) is calculated by relative frequency estimates as follows:

$$\overleftarrow{\theta}_{r+1}(s|t) = \frac{\sum_{\substack{\mathbf{s}\in\mathcal{S} \\ \mathbf{t}\in L(\mathbf{s})}} q(\mathbf{t}|\mathbf{s}; \overleftarrow{\theta}_r) \times \sum_j \delta[s_j, s]\delta[t_j, t]}{\sum_{\substack{\mathbf{s}\in\mathcal{S} \\ \mathbf{t}\in L(\mathbf{s})}} q(\mathbf{t}|\mathbf{s}; \overleftarrow{\theta}_r) \times \sum_j \delta[t_j, t]}$$

where $\delta[x, y] = 1$ iff $x == y$, and zero otherwise. The actual implementation for Hidden Markov Models is known as the Baum-Welch or Forward-Backward algorithm (Baum et al., 1970).

2.3. Existing Bi-directional Methods

It has been observed in the SMT literature that combining the alignments estimated from the two possible directions of translation $\mathcal{S} \rightarrow \mathcal{T}$ and $\mathcal{S} \leftarrow \mathcal{T}$ improves the precision of the alignment (Och & Ney, 2003). Reconciling the alignments of the two directions of translation culminates in the method of (Zens et al., 2004). This method employs two directional translation models, each with a hidden directional alignment model and a word-to-word lexicon. The crucial observation of Zens et al., shared with our approach, is that the conditional lexicon probabilities can be computed using joint estimates (see equations 4) from counts over the alignments obtained from either translation direction. Contrary to our approach, however, Zens et al. employ *two separate* Uni-EM algorithms to construct two probabilistic directional alignments. After each iteration of these Uni-EM algorithms, each of the directional alignments is used for acquiring estimates of the joint counts for the lexicon word-pairs. These joint counts are then interpolated together leading to "symmetrized" lexicon probability estimates, which are in turn fed back into each of the separate Uni-EM algorithms. It is unclear what objective function of the data this method is optimizing. Furthermore, Zens et al. make unrealistic and unnecessary assumptions regarding the unigram counts in the two corpora.

Coming right up to date, (Liang et al., 2006) present "Alignment by Agreement": The key idea is to employ the parallel corpus $\langle \mathcal{S}, \mathcal{T} \rangle$ for the estimation of two alignments $\overleftarrow{\theta}$ and $\overrightarrow{\theta}$ (the two directions of translation) under an objective likelihood function of $\langle \mathcal{S}, \mathcal{T} \rangle$ that measures individual fit to the data as well as mutual "agreement" between these alignments:

$$\mathbf{L}(\mathcal{S}, \mathcal{T}; \overrightarrow{\theta}) \times \mathbf{L}(\mathcal{S}, \mathcal{T}; \overleftarrow{\theta}) \times \mathbf{L}(\mathcal{S}, \mathcal{T}; Agr(\overrightarrow{\theta}, \overleftarrow{\theta}))$$

where $\mathbf{L}(X; \theta) = \prod_{x\in X} p_\theta(x)$ stands for the likelihood of parallel corpus X (sentence pairs) under the model that employs alignment θ, and $Agr(a, b)$ measures the agreement between the two alignments a and b given $x \in X$ as the dot product of two probability vectors that range over all possible alignments between that pair (also called set of generalized alignments).

While the idea of agreement alignment is appealing, it is by definition not applicable in the present case as we start out from a non-parallel corpus. Furthermore, because the lexicon is large (relative to sentence length), it is computationally prohibitive to employ the same measure of agreement (such as dot product) between the two estimates of probabilities (per direction) over the subsets of the translation lexicon (the power set of the lexicon).

3. Noisy-Channel Estimators

We start out from the intuition that the independent estimation of the lexicon probabilities $\tilde{p}_{\overleftarrow{\theta}}(\mathbf{s}|\mathbf{t})$ and $\tilde{p}_{\overrightarrow{\theta}}(\mathbf{t}|\mathbf{s})$ yields empirical estimates that *do not agree* on the joint probability $p(\mathbf{s}, \mathbf{t})$, i.e.

$$\tilde{p}(\mathbf{t})\tilde{p}_{\overleftarrow{\theta}}(\mathbf{s}|\mathbf{t}) \neq \tilde{p}(\mathbf{s})\tilde{p}_{\overrightarrow{\theta}}(\mathbf{t}|\mathbf{s})$$

This inequality is expected due to the asymmetric statistics in $\mathcal{T}$ and $\mathcal{S}$, asymmetry in the translation lexicon and weak language models. We hypothesize that the notion of "agreement" between the two models can be implemented by estimation under the constraint that consensus is achieved over this joint probability. A straightforward approach would be to take the weighted sum of the final EM estimates obtained over the two translation directions (each conducted on its own):

$$\tilde{p}(\mathbf{s}, \mathbf{t}) = \lambda \tilde{p}_{\overleftarrow{\theta}}(\mathbf{s}, \mathbf{t}) + (1 - \lambda)\tilde{p}_{\overrightarrow{\theta}}(\mathbf{s}, \mathbf{t}) \quad (3)$$

where λ could be, e.g. the ratio of corpora sizes. This leads to re-estimates

$$p_{\overrightarrow{\theta}}(\mathbf{s}|\mathbf{t}) = \frac{\tilde{p}(\mathbf{s}, \mathbf{t})}{\sum_{\mathbf{s}'} \tilde{p}(\mathbf{s}', \mathbf{t})} \qquad p_{\overleftarrow{\theta}}(\mathbf{t}|\mathbf{s}) = \frac{\tilde{p}(\mathbf{s}, \mathbf{t})}{\sum_{\mathbf{t}'} \tilde{p}(\mathbf{s}, \mathbf{t}')} \quad (4)$$

While interpolating the estimates could be useful, we take a novel approach that aims at maximizing the

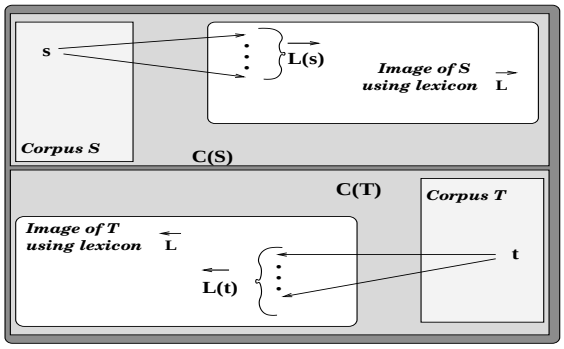

Figure 1. The concatentation of complete source and target corpora results in a single complete corpus.

joint-likelihood of the two corpora under a joint probability model $p_\theta(\mathbf{s},\mathbf{t}) = \prod_{i=1}^{n} \theta(s_i, t_i)$ which coordinates two internally hidden conditional, directed translation models that are both employing the same set of translation parameters θ. Let $p_1(\mathbf{s})$ be a language model estimated from $\mathcal{S}$ and analogously $p_2(\mathbf{t})$ from $\mathcal{T}$, we rewrite the directional translation models in terms of a single set of lexicon parameters θ:

$$\begin{aligned}\arg\max_{\mathbf{s}} p(\mathbf{s}|\mathbf{t}) &= \arg\max_{\mathbf{s}} p_1(\mathbf{s}) \frac{p_\theta(\mathbf{t},\mathbf{s})}{\sum_{\mathbf{t}'} p_\theta(\mathbf{t}',\mathbf{s})} \\ \arg\max_{\mathbf{t}} p(\mathbf{t}|\mathbf{s}) &= \arg\max_{\mathbf{t}} p_2(\mathbf{t}) \frac{p_\theta(\mathbf{t},\mathbf{s})}{\sum_{\mathbf{s}'} p_\theta(\mathbf{t},\mathbf{s}')}\end{aligned}$$

Stating the two models in terms of the same set of joint probabilities of words implies that the source and target corpora are assumed to have been generated from a single source: the joint lexicon probabilities. This allows us to state a new objective function, the Joint-Likelihood of two monolingual corpora:

$$\max_{\theta} \ \mathbf{L}(\mathcal{T};\theta,p_1,L) \times \mathbf{L}(\mathcal{S};\theta,p_2,L) \tag{5}$$

$$\begin{aligned}\mathbf{L}(X;\theta,p_k,\hat{L}) &= \prod_{\mathbf{x}\in X} \sum_{\mathbf{y}\in\hat{L}(\mathbf{x})} \mathbb{L}(\mathbf{x},\mathbf{y};\theta,p_k) \\ \mathbb{L}(\mathbf{x},\mathbf{y};\theta,p_k) &= p_k(\mathbf{y}) \frac{p_\theta(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{x}'} p_\theta(\mathbf{x}',\mathbf{y})}\end{aligned}$$

This statement of the objective function optimizes over θ the joint-likelihood of two monolingual corpora, each under its own likelihood function which involves the other corpus.

Crucially, the joint-likelihood function has the same form as the usual likelihood function with the minor difference that the multiplication ranges over two rather than one corpus (each under its own translation direction). In light of this observation we can directly obtain a Bidirectional-EM algorithm that aims at the joint-likelihood, just in the same fashion the EM is obtained from standard maximum-likelihood.

Let us define two corpora $C(\mathcal{S})$ and $C(\mathcal{T})$ (see figure 1): $C(\mathcal{S})$ is the corpus that consists of a pair $\langle \mathbf{s},\mathbf{t}\rangle$ for every sentence $\mathbf{s} \in \mathcal{S}$ and every hypothesis $\mathbf{t} \in L(\mathbf{s})$. Corpus $C(\mathcal{T})$ is defined analogously. Figure 2 shows the Bi-EM algorithm,

E-step$_r$:

$$\forall \langle \mathbf{s},\mathbf{t}\rangle \in C(\mathcal{T}):\ q^1(\mathbf{s},\mathbf{t};\theta_r) := p_1(\mathbf{s}) \textstyle\prod_{i=1}^{n} \frac{\theta_r(s_i,t_i)}{\sum_t \theta_r(s_i,t)}$$

$$\forall \langle \mathbf{s},\mathbf{t}\rangle \in C(\mathcal{S}):\ q^2(\mathbf{s},\mathbf{t};\theta_r) := p_2(\mathbf{t}) \textstyle\prod_{i=1}^{n} \frac{\theta_r(s_i,t_i)}{\sum_s \theta_r(s,t_i)}$$

M-step$_r$: maximize over θ to obtain θ_{r+1}

$$\theta_{r+1} := \arg\max_{\theta} \sum_{\mathbf{s},\mathbf{t}\in C(\mathcal{T})} \overbrace{\frac{q^1(\mathbf{s},\mathbf{t};\theta_r)}{Z^1(\mathbf{s};\theta_r)} \log \mathbb{L}(\mathbf{s},\mathbf{t};\theta,p_1)}^{A_r(\mathbf{s},\mathbf{t};\theta)} + \sum_{\mathbf{s},\mathbf{t}\in C(\mathcal{S})} \overbrace{\frac{q^2(\mathbf{s},\mathbf{t};\theta_r)}{Z^2(\mathbf{t};\theta_r)} \log \mathbb{L}(\mathbf{t},\mathbf{s};\theta,p_2)}^{B_r(\mathbf{s},\mathbf{t};\theta)}$$

$$\mathbb{L}(\mathbf{x},\mathbf{y};\theta,p) = p(\mathbf{x}) \textstyle\prod_{i=1}^{n} \frac{\theta(x_i,y_i)}{\sum_y \theta(x_i,y)}$$

Figure 2. Bi-EM algorithm

where $Z^1(\mathbf{t};\theta_r) = \sum_{\mathbf{s}\in L(\mathbf{t})} q^1(\mathbf{s},\mathbf{t};\theta_r)$ and $Z^2(\mathbf{s};\theta_r) = \sum_{\mathbf{t}\in L(\mathbf{s})} q^2(\mathbf{s},\mathbf{t};\theta_r)$ are unigram count estimates.

The sum of the two sums in the M-step can be rearranged into a single sum if we precompute a single (complete) corpus $\mathfrak{C}_r$ that concatenates $C(\mathcal{S})$ with $C(\mathcal{T})$ and stores the expected frequencies ($A_r(\mathbf{s},\mathbf{t};\theta)$ or $B_r(\mathbf{s},\mathbf{t};\theta)$) with each pair as

$$\log \mathfrak{freq}_r(\mathbf{s},\mathbf{t};\theta) = \begin{cases} A_r(\mathbf{s},\mathbf{t};\theta) & \langle \mathbf{s},\mathbf{t}\rangle \in C(\mathcal{T}) \\ B_r(\mathbf{s},\mathbf{t};\theta) & \langle \mathbf{s},\mathbf{t}\rangle \in C(\mathcal{S}) \end{cases}$$

The M-step becomes the M-step of a standard EM algorithm:

$$\theta_{r+1} := \arg\max_{\theta} \sum_{\langle \mathbf{s},\mathbf{t}\rangle \in \mathfrak{C}_r} \log \mathfrak{freq}_r(\mathbf{s},\mathbf{t};\theta)$$

Hence, the Bi-EM inherits the properties of the common (Uni-directional/Uni-) EM algorithm, including convergence and a guarantee of a choice of θ that will not decrease the joint-likelihood after each iteration.

The actual update formula for the Bi-EM is:

$$q(\mathbf{s},\mathbf{t};\theta_r) = \begin{cases} q^1(\mathbf{s},\mathbf{t};\theta_r) & \langle \mathbf{s},\mathbf{t}\rangle \in C(\mathcal{T}) \\ q^2(\mathbf{s},\mathbf{t};\theta_r) & \langle \mathbf{s},\mathbf{t}\rangle \in C(\mathcal{S}) \end{cases}$$

$$\theta_{r+1}(s,t) = \frac{\sum_{\langle \mathbf{s},\mathbf{t}\rangle \in \mathfrak{C}_r} q(\mathbf{s},\mathbf{t};\theta_r) \times \sum_j \delta[s_j,s]\delta[t_j,t]}{\sum_{\langle \mathbf{s},\mathbf{t}\rangle \in \mathfrak{C}_r} q(\mathbf{s},\mathbf{t};\theta_r)}$$

Note that the Bi-EM takes only twice as much training time as the Uni-EM.

4. Implementation Detail

The core of both the Uni-EM estimation methods (Koehn & Knight, 2000) and the present Bi-EM estimator is the Baum-Welch algorithm (Baum et al., 1970) for Hidden Markov Models (HMMs), which is known to be an EM algorithm (Dempster et al., 1977). This algorithm in its most general form employs the Forward-Backward calculations to update expected counts of transition (language model) and emission (lexicon) probabilities. In our setting we fix the language model (transition) estimates and reestimate only the lexicon (emission) probabilities. This is because language models can be readily constructed from large monolingual data and there is no reason to reestimate them.

For the generation of the language models we used the CMU-Cambridge Toolkit (Clarkson & Rosenfeld, 1997), employing a first order Markov model. For the Baum-Welch algorithm, we implemented our own (Java) software package. Our software package[3] implements both the Uni- and Bi-EM algorithms. For POS-tagging we employ the TnT tagger (Brants, 2000) which works with a 2nd-order HMM over POS tags and individual lexical (word-tag) probabilities.

5. Application I: Translation

Following (Koehn & Knight, 2000), our experiments are on translating noun sequences extracted from corpus sentences. As an absolute baseline we employ a translation model that assumes uniform lexicon probabilities (called *'LM' method*). The actual baseline, however, is the standard EM (Koehn & Knight, 2000) (subsequently called *Uni-EM* – Unidirectional EM). We compare these baselines to the present *Bi-EM* algorithm (section 3).

During training, the input to the estimation methods consists of a non-parallel English-German corpus pair and an ambiguous lexicon containing up to seven German translations for every English word.[4] We initialize the lexicon parameters with a uniform distribution both for Uni- and Bi-EM.

For evaluation purposes, we embed the lexicon estimates within a simple word-to-word translation system (section 2.2), and evaluate the translation result against the translations available in a given parallel corpus. As (Koehn & Knight, 2000), we use German-to-English translation. As a test corpus we use 5106 word translation pairs from 1850 noun sequences extracted from an equal number of sentences from the de-news[5], which have been aligned down to the word level. We measure *accuracy*, the fraction of words whose translation matches the word used in the bi-text. In addition, we also provide the *BLEU* scores (Papineni et al., 2001) as an additional measure of translation quality.

5.1. Effect of Domain Mismatch

The different estimators operate under domain- and/or genre-mismatch between (1) source corpus, (2) target corpus, (3) lexicon, and (4) test corpus. We fix the lexicon and the test corpus throughout all experiments. Because the Bi-EM aims at the joint-likelihood of two corpora, a question may arise as to whether weakening the relatedness (in domain and/or genre) of the two corpora will affect the performance of Bi-EM relative to Uni-EM.

Highly related The two corpora here consist of noun sequences from two *non-overlapping* sections of the Europarl (Koehn, 2005) parallel corpus (English-German). The baseline system using the LM method (uniform lexicon probabilities) achieves an accuracy of 63.11% (BLEU score 0.2372). The following table list Uni- vs. Bi-EM results:

#sentences	Uni-EM	Bi-EM
40K	72.01% (0.3896)	76.19% (0.4394)
75K	74.13% (0.4242)	77.34% (0.4660)
100K	74.99% (0.4300)	77.78% (0.4714)

Compared against the baseline (63.11% for the 'LM' method) these numbers improve by up to 15% (or in fact 40% error reduction). Bi-EM clearly outperforms the standard Uni-EM. It is evident from the results that the improved accuracy of the Bi-EM does not come from utilizing more data. Bi-EM trained on 40,000 English and the same amount of German

[3] `http://staff.science.uva.nl/~mmylonak`

[4] The lexicon was obtained by automatic word alignment of the Europarl corpus.

[5] `http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/de-news/`

sentences significantly outperforms Uni-EM trained on 100,000 English sentences (and a German language model). This is a strong indication that the Joint-Likelihood is a better objective function than the likelihood of a single corpus.

Less related We use as training data newspaper text from the Gigaword (English) and from the European Language Newspaper Text (German), utilizing news stories coming from the same agencies and published during the same period (Associated Press, Agence France-Presse, May 1994-December 1995). Unlike different sections of Europarl, this pair of corpora concerns news texts that originate from non-parallel sources and are in two different languages. We estimate translation probabilities using Uni-EM and Bi-EM, training with 100K sentences per language used:

#sentences	Uni-EM	Bi-EM
100K	70.29% (0.3610)	72.80% (0.3809)

We notice again that the Bi-EM helps produce significantly more accurate translations. Interestingly, training Bi-EM on 100K sentence still gives better results that Uni-EM trained on 200K sentences (Uni-EM with 200K = 72.08% (0.3737)).

Distantly related We also trained on a pair of distantly related corpora. These are the newspaper text from Gigaword (English) and the parliament proceedings from Europarl (German):

#sentences	Uni-EM	Bi-EM
100K	68.90% (0.3110)	70.98% (0.3303)

Bi-EM is still able to produce estimates that give more accurate translations than Uni-EM. Furthermore, Bi-EM trained on 100K sentences outperforms Uni-EM trained on 200K sentences (Uni-EM on 200K = 70.23% (0.3215)).

5.2. Smaller Target Language Data

We employ the corpora of section 5.1, varying this time the amount of training sentences from the target language (English), while maintaining a fixed training corpus of 100K German sentences (source). Figure 3 shows the average accuracies of Bi-EM as function of target corpus increase. Note that the zero point refers to the Bi-EM trained on target corpus of size zero, which is equivalent to the Uni-EM. Interestingly, 81% of the accuracy increase of Bi-EM relative to Uni-EM is already obtained by using only 25K sentences, 77.32% (0.4542). These accuracies are averages over 3 different non-overlapping sets of 25K English sentences.

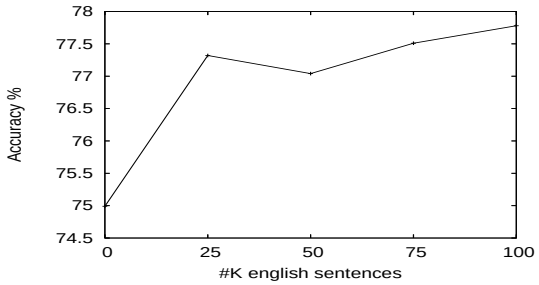

Figure 3. Bi-EM accuracy as target corpus size increases

6. Application II: Adapting Taggers

Part-of-Speech (POS) tagging is the task of classifying every word in a text into one POS category (e.g., verb, noun). Many machine learning techniques have been applied to POS tagging, including HMMs, Conditional Random Fields, Support Vector Machines, Memory-Based Learning, just to name a few (Ratnaparkhi, 1996; Daelemans et al., 1996; Brants, 2000; Lafferty et al., 2001).

Here we focus on the POS tagging of transcripts of a spoken Levantine Arabic dialect. Unlike Modern Standard Arabic (MSA), Arabic dialects are spoken but rarely ever written, which makes it virtually impossible to obtain MSA-dialect parallel corpora (see (Rambow et al., 2005)). Available is a manually tagged MSA corpus (approx. 564K words) (Maamouri et al., 2004) and a *tiny*, manually created translation lexicon [6] that maps words between Levantine and MSA. Also available is a small Levantine corpus (approx. 31K words) consisting of two splits (18157 and 12238 words resp.). The task here is to utilize the MSA tagged corpus in order to automatically POS tag the Levantine side using only unannotated Levantine sentences for training and the lexicon for translation.

We embed the MSA POS tagger and MSA-Levantine lexicon in the noisy-channel approach. Let $\mathbf{m} = m_1 \dots m_n$ be an MSA sentence and $\mathbf{l} = l_1 \dots l_n$ be a Levantine sentence. On the MSA side we have a POS tag sequence $\mathbf{t} = t_1 \dots t_n$ associated with $\mathbf{m}$. We have two directions for the noisy channel:

$$P(\mathbf{m}, \mathbf{t}, \mathbf{l}) = P(\mathbf{m}, \mathbf{t}) P(\mathbf{l} \mid \mathbf{m}, \mathbf{t}) \quad (6)$$

$$P(\mathbf{m}, \mathbf{t}, \mathbf{l}) = P(\mathbf{l}) P(\mathbf{m}, \mathbf{t} \mid \mathbf{l}), \quad (7)$$

[6] Originating from JHU 2005 summer workshop `http://www.clsp.jhu.edu/ws2005/`. The lexicon has 321 entries with on average approx. 1.5 Levantine words per MSA word; If averaged over ambiguous MSA words only, the ambiguity rises to 3.

Table 1. Adapting MSA POS tagger to Levantine

Adaptation	Training Data	Accuracy
None	MSA only	70.48%
Uni-EM	MSA-to-Lev	75.93%
Uni-EM	Lev-to-MSA	77.88%
Bi-EM	MSA-and-Lev	78.25%

where $P(\mathbf{m}, \mathbf{t})$ is an MSA POS tagger, $P(\mathbf{l})$ is a Levantine Language Model, and the other two terms are channel models involving the translation lexicon in the two directions. The 2^{nd}-order HMM MSA POS tagger and the Levantine language model are both standard:[7].

$$P(\mathbf{m}, \mathbf{t}) = \prod_{i=1}^{n} P(t_i|t_{i-2}, t_{i-1})P(m_i|t_i) \quad (8)$$

$$P(\mathbf{l}) = \prod_{i=1}^{n} P(l_i|l_{i-1}) \quad (9)$$

For equation 8, we train an off-the-shelf HMM POS tagger (Brants, 2000) on the MSA data (accuracy 95.07 over 66K words test-set).

We make two strong assumptions (1) The Levantine POS tagger differs from the MSA POS tagger only in the lexical model, and (2) When a Levantine word translates into an MSA word-tag pair, the POS tag remains the same. The latter means that we extend the MSA-Levantine lexicon from pairs $\langle m, l\rangle$ into triples $\langle m, t, l\rangle$, where t is any of the POS tags that co-occur with word m in the tagged MSA corpus. A word found in both corpora but not in the lexicon is mapped to itself, and a word found in the Levantine but not in the MSA corpus nor in the lexicon is mapped to all open category POS tags.

For the two Uni-EM versions, the channel probability employs the probabilistic lexicon in two directions $P(\mathbf{l}|\mathbf{m}, \mathbf{t}) = \prod_{i=1}^{n} \overrightarrow{\theta}(l_i|m_i, t_i)$ and $P(\mathbf{m}, \mathbf{t}|\mathbf{l}) = \prod_{i=1}^{n} \overleftarrow{\theta}(m_i, t_i|l_i)$. For the Bi-EM we assume one (non-directional/joint) set of parameters $\theta(m, t, l)$ that underlies the two directional/conditional parameters as done within the translation task (section 5). The estimate $\theta(m, t, l)$ is converted into a Levantine lexical model: $P(l|t) = \frac{\sum_m \theta(m,t,l)}{\sum_{\langle m,l\rangle} \theta(m,t,l)}$. This lexical model is used together with the 2^{nd}-order Markov Model over POS tags (trained on the MSA corpus) as a Levantine POS tagger.

Table 1 exhibits the results of the various POS taggers on the Levantine data averaged over two splits (the two Lev parts). The first row is the original MSA trained POS tagger (70.48% accuracy = percentage of correctly tagged test words). The second and third rows correspond each to an adapted MSA POS tagger using the Uni-EM estimates from either translation direction. Depending on direction, the Uni-EM achieves 18-25% less errors relative to the unadapted tagger. The Bi-EM adapted POS tagger (last row) commits 2-10% less errors than the Uni-EM directions (or about 27.5% less errors than the MSA POS tagger).

Note that we have *not* included any external knowledge. In (Rambow et al., 2005), manual adaptation combined with EM leads to 77-78% accuracy on a *modified* version[8] of the Levantine data. On that test-material, our experiments show that the Bi-EM scores 82.30% accuracy (averaged over two splits).

We think that two factors contribute to the fact that the Bi-EM improves over Uni-EM: (1) It combines statistics from the MSA POS tagger (one direction) with statistics from the Levantine language model (another direction), and (2) Because the lexicon is asymmetric, Uni-EM updates only those entries used in the assumed direction, whereas Bi-EM updates the lexicon entries used in both directions.

7. Conclusions

This paper aims at improved channel estimates from data at both ends of the noisy-channel. We presented a Joint Maximum-Likelihood approach and extended the EM algorithm into a bi-directional EM for unsupervised estimation. We exemplified the utility of Bi-EM on two tasks: translation by lexicon probability estimates and adaptation of a POS tagger from a resource-rich to a resource-poor language. Bi-EM delivers better results than the standard EM regardless of mismatch in domain or genre between the source and target corpora.

In future work we aim at utilizing the Bi-EM for porting more linguistic processing tools from a resource-rich to a resource-poor language in cases where there exist no parallel corpora. We also think that the Bi-EM could be useful in statistical machine translation, in particular for obtaining improved translation model estimates. Whenever the channel model (lexicon) is asymmetric and/or the language models are weak, it makes more sense to employ Bi-EM than standard (Uni-directional) EM for noisy-channel applications.

[7]For brevity, any symbol x_j where $j \leq 0$ is assumed to be the unique start symbol of a sentence.

[8]Clitics are marked with disambiguating symbols.

Acknowledgements

Preliminary Bi-EM versions were explored by the 2nd and 3rd authors, together with Carol Nichols, during 2005 JHU Summer Language Engineering Workshop. We thank the JHU organizers for the opportunity, the workshop participants for discussions and data, the ICML reviewers for comments, Andy Way and Hermann Ney for pointers to relevant literature, and Aspasia Beneti and Isaac Esteban for help with preliminary experiments. The first author is supported by a NUFFIC HSP Huygens scholarship HSP-HP.06/940-G, and the second author by NWO grant number 639.022.604.

References

Bahl, L. R., Jelinek, F., & Mercer, R. L. (1990). *A maximum likelihood approach to continuous speech recognition*, 308–319. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Baum, L., Peterie, T., Souled, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, *41*, 164–171.

Brants, T. (2000). Tnt: a statistical part-of-speech tagger. *Proceedings of the sixth conference on Applied natural language processing* (pp. 224–231). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Brown, P., Cocke, J., Pietra, S. D., Jelinek, F., Mercer, R., & Roossin, P. (1988). A statistical approach to language translation. *COLING-88*.

Clarkson, P., & Rosenfeld, R. (1997). Statistical language modeling using the cmu-cambridge toolkit. *Proceedings ESCA Eurospeech*.

Daelemans, W., Zavrel, J., Berck, P., & Gillis, S. (1996). Mbt: A memory-based part of speech tagger generator. *Proceedings of the fourth Workshop on Very Large Corpora (ACL SIGDAT)* (pp. 14–27). Copenhagen, Denmark.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. *MT Summit*.

Koehn, P., & Knight, K. (2000). Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. *AAAI/IAAI*.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*. Edmonton, Canada.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning* (pp. 282–289). Morgan Kaufmann, San Francisco, CA.

Liang, P., Taskar, B., & Klein, D. (2006). Alignment by agreement. *Proceedings of the Human Language Technology Conference (HLT-NAACL 2006)*. New York.

Maamouri, M., Bies, A., Buckwalter, T., & Mekki, W. (2004). The penn arabic treebank: Building a large-scale annotated arabic corpus. *Proceedings of NEMLAR 2004*..

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*, 19–51.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 311–318). Morristown, NJ, USA: Association for Computational Linguistics.

Rambow, O., Chiang, D., Diab, M., Habash, N., Hwa, R., Sima'an, K., Lacey, V., Levy, R., Nichols, C., & Shareef, S. (2005). *Parsing arabic dialects* (Technical Report). Johns Hopkins University 2005 Summer Workshop on Language Engineering.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana: University of Illinois Press.

Zens, R., Matusov, E., & Ney, H. (2004). Improved word alignment using a symmetric lexicon model. *Proceedings of the 20th International Conference on Computational Linguistics (CoLing)* (pp. 36–42). Geneva, Switzerland.